

1. RESUMO: você deverá produzir uma implementação de DEQUE ILIMITADO(A) em C++ atendendo às especificações abaixo.

2. ENTREGA: o seu código-fonte deverá ser entregue por meio do endereço

<http://dc.ufc.br/~pablo/2019-2/ed/trab1/>

até o dia 16/09/2019. Cada código submetido será avaliado automaticamente e obterá nota de 0 a 0,5. Caso a avaliação aponte um erro no seu código, você poderá fazer nova submissão, desde que no prazo.

3. ESPECIFICAÇÕES:

3.1. Um(a) DEQUE ("double-ended queue") é uma generalização da fila, na qual inserções e remoções podem acontecer em ambas as extremidades. Todos os elementos armazenados constituem uma única sequência, de forma que elementos inseridos de um lado podem ser removidos do outro. Neste trabalho, as extremidades da sequência serão chamadas de "esquerda" e "direita" (ao invés de "início" e "fim", por exemplo).

3.2. Usando templates, você deverá fornecer uma definição paramétrica de deque, de que forma que, para cada tipo "T", seja definível o tipo "Deque<T>".

3.3. Você deverá definir um molde

```
template <typename T> bool inicializar (Deque<T> &D)
```

que inicialize o deque D usando um vetor de tamanho 1. Como de costume, o valor retornado deverá ser "true" se e somente se houver erro de alocação.

```
3.4. template <typename T> void terminar (Deque<T> &D)
```

Essa função (molde de função) deverá desalocar o vetor usado por D.

```
3.5. template <typename T> bool vazio (Deque<T> &D)
```

Essa função deve retornar "true" sse (se e somente se) D estiver vazio.

```
3.6. template <typename T> bool inserir_esq (Deque<T> &D, T e)
```

Essa função deverá inserir o elemento "e" na extremidade esquerda de D. Caso o vetor utilizado por D esteja cheio, deverá ser alocado um com o dobro do tamanho para substituí-lo. A função deve retornar "true" sse houver erro na inserção (o que só será o caso se for necessária uma alocação de vetor maior e se não houver memória suficiente para isso).

```
3.7. template <typename T> bool inserir_dir (Deque<T> &D, T e)
```

Deverá inserir "e" na extremidade direita de D, analogamente à função anterior.

```
3.8. template <typename T> T remover_esq (Deque<T> &D)
     template <typename T> T remover_dir (Deque<T> &D)
```

Deverá remover e retornar o elemento da extremidade esquerda/direita de D, pressupondo que D não está vazio.

3.9. Você poderá definir funções auxiliares, que sejam chamadas pelas acima.